# RAGServe:Fast Quality-Aware RAG Systems with Configuration Adaptation

Siddhant Ray, Rui Pan*, Zhuohan Gu, Kuntai Du, Shaoting Feng, Ganesh Ananthanarayanan† ,
Ravi Netravali*, Junchen Jiang

*University of Chicago †Microsoft Research *Princeton University*

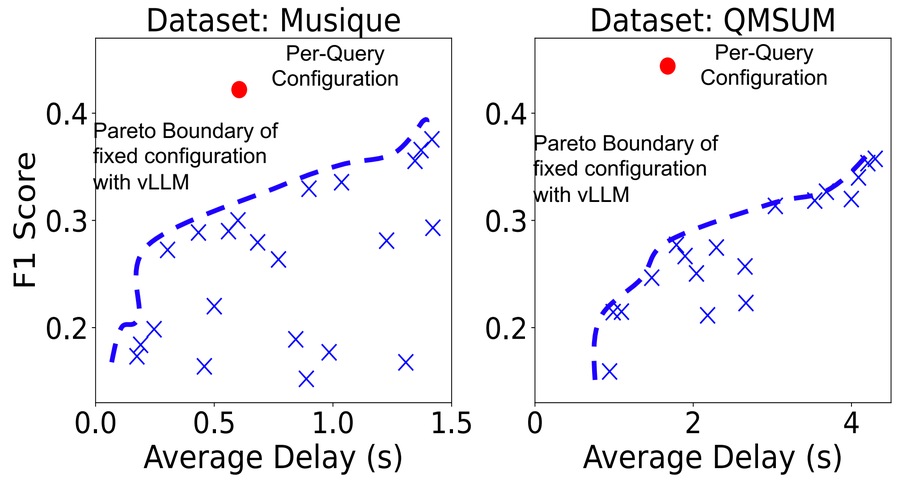
## Introduction

### Motivation

- **RAG queries have an associated *RAG Configuration***
  - *E.g., how to combine and how much data to input for a query*
- **Configurations *simultaneously* affect quality and response delay**
  - *E.g., retrieving too many chunks for simple RAG queries unnecessarily inflates delay without increasing quality*

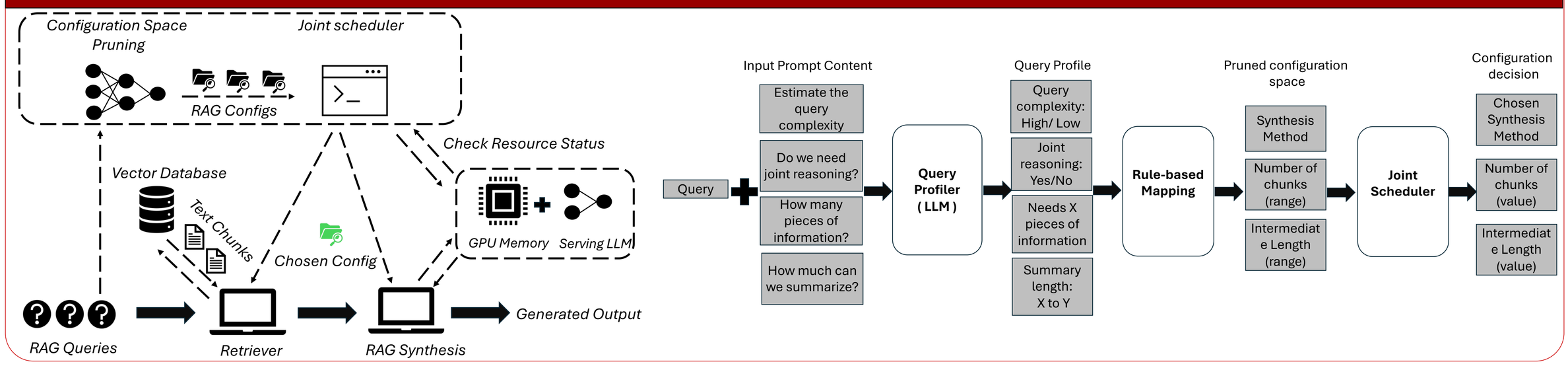
### Challenges

- **Tuning configurations quickly hits a prohibitive combinatorial space**
- **RAG configuration should be tuned jointly with scheduling**
  - *Need to achieve maximum quality at minimum delay*




Per-query configuration can achieve **10% higher** quality and **3x lower** delay compared to static configurations

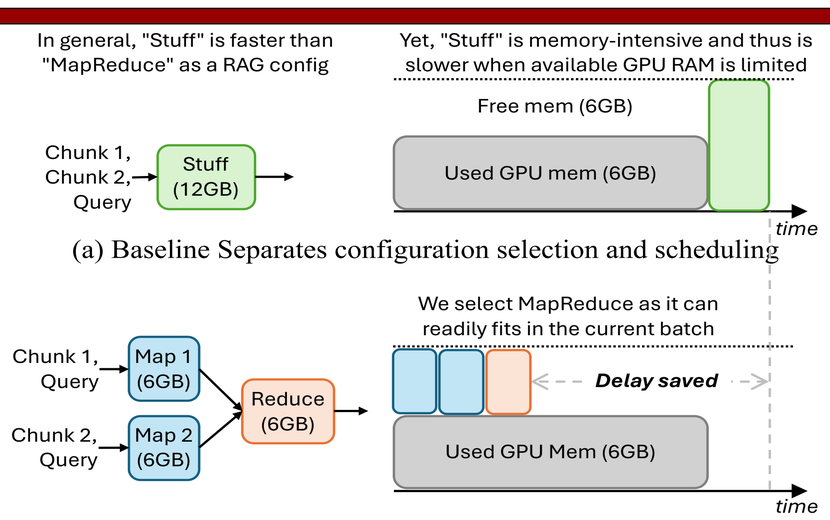
## Overview: RAG Controller with LLM Profiling and Configuration Space Pruning

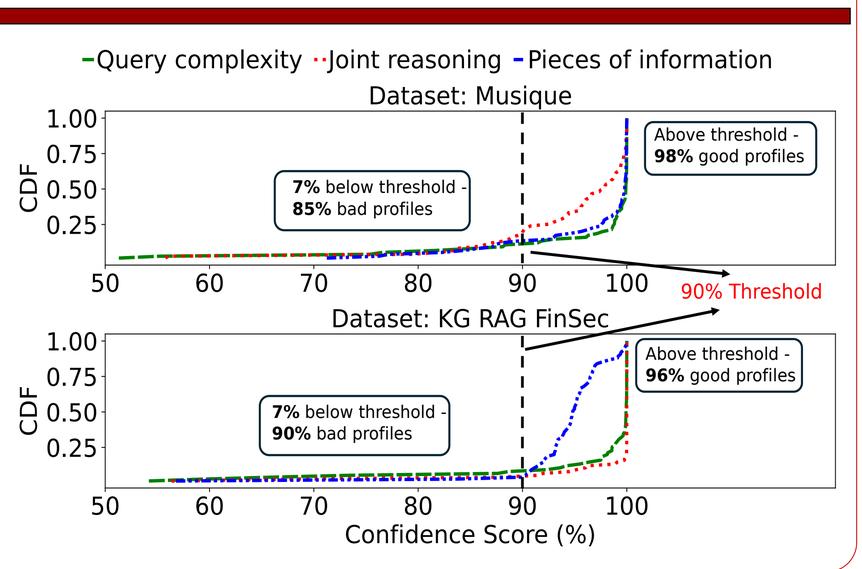



## Design: Rule-Based Mapping, Joint Scheduling and Fallback Schemes

**Input:** *Query complexity, Joint reasoning required*
**Input:** *Pieces of information , Summarization length range*
**Result:** synthesis_method, num_chunks, intermediate_length

```
if Joint reasoning required == "no" then
    synthesis_method = map_rerank
else
    if Query complexity == "low" then
        synthesis_method = stuff
    else
        synthesis_method = stuff, map_reduce
num_chunks = [Pieces of information , 3× Pieces of information]
intermediate_length_range = Summarization length range
```




## Evaluation: On Real World RAG Question-Answering Datasets